Amendments to the Claims:

This listing of claims will replace all prior versions, and listing, of claims in the application:

1.   (Currently Amended) A method of redirecting video text data in a computer network, the method comprising:

(a)   during a timer interrupt for a first computer, selecting a line of video text data from among a plurality of lines of video text data stored in a video text data memory as a current line of video text data;

(b)   calculating a checksum for the current line of video text data;

(c)   comparing the calculated checksum to a previously saved checksum for the current line of video text data;

(d)   if the calculated checksum is different than the previously saved checksum for the current line of video text data, then determining that the current line of video text data in the video text data memory has changed;

(e)   saving the calculated checksum in a memory location after determining that the current line of video text data in the video text data memory has changed;

(f)   storing the current line of video text data in a shared memory in response to determining that the current line of video text data in the video text data memory has changed;

(g)   selecting the next line of video text data in the video text data memory as the current line of video data; ~~and~~

(h)   repeating the operations (b)-(g) for each of the plurality of lines of video text data stored in the video text data memory; and

(i)   sending only the changed video text data from the first computer to a second computer in the computer network.

2.   (Cancelled)

3.   (Previously Presented)   The method of claim 1, wherein calculating a checksum for the current line of video text data comprises calculating a unique value from at least one of:

> an attribute of text in the line of video text data; and
>
> a line position of the text in the line of video text data.

4.   (Cancelled)

5.   (Previously Presented)   The method of claim 1 further comprising determining whether the shared memory is full prior to storing the current line of changed video text data in the memory.

6.   (Previously Presented)   The method of claim 1, wherein sending only the changed video text data from the first computer to a second computer in the computer network comprises sending each changed line of video text data stored in the shared memory during the timer interrupt from the first computer to the second computer in the computer network.

7.   (Previously Presented)   The method of claim 6, wherein sending each changed line of video text data stored in the shared memory during the timer interrupt from the first computer to the second computer in the computer network comprises sending each changed line of video text data stored in the shared memory from the first computer to the second computer in the computer network in response to determining that the memory is full.

8.   (Original)   The method of claim 3, wherein the text comprises a plurality of characters in the line of video text data.

9.   (Cancelled)

10. (Previously Presented) The method of claim 1, wherein the shared memory comprises a send buffer and a receive buffer.

11. (Previously Presented) The method of claim 10, further comprising:

during the timer interrupt, checking the receive buffer in the first computer for keyboard data from the second computer; and

if the receive buffer contains the keyboard data, then storing the keyboard data in the first computer.

12. (Currently Amended) A computer system for redirecting video text data, comprising:

a controller for sending and receiving data;

a memory for storing executable program code;

a video buffer for storing video text data while displayed on a display unit; and

a processor, functionally coupled to the controller, the memory, and the video text data memory, the processor being responsive to computer-executable instructions contained in the program code and operative to:

(a) during the system timer interrupt, select a line of video text data from among a plurality of lines of video text data stored in the video text data memory as a current line of video text data;

(b) calculate a checksum for the current line of video text data;

(c) compare the calculated checksum to a previously saved checksum for the current line of video text data;

(d) if the calculated checksum is different than the previously saved checksum for the current line of video text data, then determine that the current line of video text data in the video text data memory has changed;

(e) save the calculated checksum in a memory location in the computer system after determining that the current line of video text data in the video text data memory has changed; and

-11-

(f)   store the current line of video text data in the shared memory in the controller in response to determining that the current line of video text data in the video text data memory has changed;

(g)   select the next line of video text data in the video text data memory as the current line of video text data;

(h)   repeat the operations (b)-(g) for each of the plurality of lines of video text data stored in the video text data memory; and

(i)   send the changed video text data from the controller to a video display device.

13.   (Cancelled)

14.   (Previously Presented)   The computer system of claim 12, wherein the processor in calculating a checksum for the current line of video text data, is operative to calculate a unique value from at least one of:

an attribute of text in the line of video text data; and

a line position of the text in the line of video text data.

15.   (Cancelled)

16.   (Previously Presented)   The computer system of claim 12, wherein the processor is further operative to determine whether the shared memory is full prior to storing the current line of changed video text data.

17.   (Cancelled)

18.   (Original)   The computer system of claim 12, wherein the memory is a random access memory.

19. (Original) The computer system of claim 12, wherein the executable program code comprises a basic input/output system.

20. (Original) The computer system of claim 12, wherein the executable program code is stored in an extended BIOS data area in the memory.

21. (Previously Presented) The computer system of claim 12, wherein the shared memory comprises a send buffer and a receive buffer.

22. (Previously Presented) The computer system of claim 21, wherein the processor is further operative to:

during the system timer interrupt, check the receive buffer for keyboard data sent from a keyboard controller; and

if the receive buffer contains the keyboard data, then store the keyboard data in the computer system.

23. (Previously Presented) A computer-readable medium having computer-executable instructions for performing a method of redirecting video text data in a computer network, comprising:

(a) during a system timer interrupt of a first computer in the computer network, selecting a line of video text data from among a plurality of lines of video text data stored in a video text data memory as a current line of video text data;

(b) calculating a checksum for the current line of video text data;

(c) comparing the calculated checksum to a previously saved checksum for the current line of video text data;

(d) if the calculated checksum is different than the previously saved checksum for the current line of video text data, then determining that the current line of video text data in the video text data memory has changed;

(e)   saving the calculated checksum in a memory location after determining that the current line of video text data in the video text data memory has changed;

(f)   storing the current line of video text data in a shared memory in response to determining that the current line of video text data in the video text data memory has changed;

(g)   selecting the next line of video text data in the video text data memory as the current line of video text data;

(h)   repeating the operations (b)-(g) for each of the plurality of lines of video text data stored in the video text data memory; and

(i)   sending each changed line of video text data stored in the shared memory during the system timer interrupt from the first computer to the second computer in the computer network.

24.   (Cancelled)